

Fare Evasion in Transit Networks

José R. Correa¹, Tobias Harks², Vincent J.C. Kreuzen², and Jannik Matuschke³

¹ Departamento de Ingeniería Industrial, Universidad de Chile

² School of Business and Economics, Maastricht University

³ Institut für Mathematik, Technische Universität Berlin

Abstract. Public transit systems in urban areas usually require large state subsidies, primarily due to high fare evasion rates. In this paper, we study new models for optimizing fare inspection strategies in transit networks based on bilevel programming. In the first level, the leader (the network operator) determines probabilities for inspecting passengers at different locations, while in the second level, the followers (the fare-evading passengers) respond by optimizing their routes given the inspection probabilities and travel times. To model the followers' behavior we study both a non-adaptive variant, in which passengers select a path a priori and continue along it throughout their journey, and an adaptive variant, in which they gain information along the way and use it to update their route. For these problems, which are interesting in their own right, we design exact and approximation algorithms and we prove a tight bound of $3/4$ on the ratio of the optimal cost between adaptive and non-adaptive strategies. For the leader's optimization problem, we study a fixed-fare and a flexible-fare variant, where ticket prices may or may not be set at the operator's will. For the latter variant, we design an LP based approximation algorithm. Finally, using a local search procedure that shifts inspection probabilities within an initially determined support set, we perform an extensive computational study for all variants of the problem on instances of the Dutch railway and the Amsterdam subway network. This study reveals that our solutions are within 95% of theoretical upper bounds drawn from the LP relaxation.

1 Introduction

Fare evasion in transit systems causes significant losses to society. Recent studies revealed an annual loss of 70 million pounds for London's transit system [21] and fare evasion rates of nearly 20% in Santiago's transit system, which is subsidised with over 700 million dollars each year [19]. As the installation of physical ticket barriers is not always possible, cost-efficient, or desirable, many transit systems, such as the Dutch and German railway and subway networks and the London bus system, rely on the honesty of customers and proper controlling of valid tickets *on board*. Travelers who are caught without a ticket have to pay a fine, which is significantly larger than the ticket price.

In this paper, we study the optimization of fare inspection strategies in transit systems taking into account realistic models for the passenger's reaction. Our

models are based on a *bilevel optimization problem* (or Stackelberg game) on the network. In the first level, the leader, who strives to maximize the revenue from ticket sales and collected fines, determines for each edge the probability of controlling passing passengers, representing the frequency of inspections on that edge. In light of the limited number of inspectors available, we assume a global budget constraint on the sum of these probabilities. Given the inspection probabilities on edges, we assume passengers to act strategically and decide on whether or not to buy a ticket and along which path to travel based solely on their perceived cost. If a passenger chooses not to buy a ticket, her cost is thus the sum of the path length (expressed in monetary units) and the expected fine to be paid (which depends on the inspection probabilities along the path); on the other hand, if she buys a ticket, her cost is defined as the cost of the cheapest path considering both ticket price and distance measured in monetary units.

We perform a complete study differentiating between two different possibilities for the passengers' behavioral assumption as well as two different settings for the leader's decision problem. From the followers side, we consider both the case in which they are *adaptive* and a *non-adaptive*. In the latter, each passenger chooses a path at the beginning of her journey and continues along it, independent of whether or not she encounters an inspector. In the adaptive version, passengers adapt their behavior and consequently, when caught without a ticket, they continue on their shortest path only considering distance (since the fine includes a ticket to finish the trip). For the leader, we also study two different settings. In the *fixed-fare* setting, we assume that ticket prices are fixed a priori, e.g., by governmental regulations, and the leader only sets the inspection probabilities subject to a budget constraint. In the *flexible-fare* setting, we assume that the leader can additionally determine ticket prices.

Our Contribution and Structure of the Paper. The main goal of this work is to provide a comprehensive study of fare evasion and fare inspection problems in transit networks. As mentioned above, we consider four versions of the problem by varying on whether the followers are adaptive or not, and whether the ticket prices are fixed or flexible. From a methodological viewpoint we tackle the problem from different angles, designing polynomial time algorithms and approximation schemes for the followers' problems, studying approximation algorithms and LP relaxations for the leader's problem, and using a local search heuristic to obtain high quality solutions for real-world instances.

After establishing the precise model in Section 2, we study the two variants of the followers' minimization problem in Section 3. These are natural extensions of the classic shortest path problem and exhibit interesting properties on their own which go beyond the particular application in this work. For the non-adaptive variant, we design a fully polynomial time approximation scheme for general network topologies, exploiting similarities between this problem and the restricted shortest path problem (RSP). Interestingly we also obtain an exact polynomial time algorithm for series-parallel graphs, in sharp contrast to RSP, which is still *NP*-hard in these graphs. For adaptive followers, we obtain an exact polynomial

time algorithm for general graphs using an optimal substructure property of the problem. Surprisingly, we further show that for arbitrary probability distributions, the optimal solution to the non-adaptive followers’ minimization problem is at most a factor $4/3$ away from the optimal solution to the adaptive variant, and we also show that this bound is tight.

In Section 4, we turn our attention to the leader’s maximization problem and prove that all four variants of this problem are strongly NP-hard. We then present an LP relaxation that yields a valid upper bound on the achievable profit for all four variants, and also obtain a $(1 - 1/e)$ -approximation algorithm for the variant involving flexible ticket prices and non-adaptive followers. Combining this with the worst-case gap from the preceding section yields a $\frac{3}{4}(1 - 1/e)$ -approximation for the variant with adaptive followers. Finally, we develop a local search procedure that shifts inspection probabilities within an initially determined support set of edges. As candidate support sets, we use solutions from an LP relaxation, a minimum multicut, and a related mixed integer program [1].

In Section 5, we demonstrate the applicability of our local search approach for all four problem variants by conducting an extensive computational study comprising a total of 5600 instances based on the networks of the Dutch railway system, the Amsterdam metro system, and randomly generated graphs. Our study reveals that the objective values of the computed solutions are within 95% of the calculated upper bound on average.

Related Work. Bilevel network pricing problems have been considered extensively in the literature; see [6, 7, 15]. In these works, the leader prices edges of some subgraph and the followers choose shortest paths with respect to edge costs defined as the sum of travel costs and prices. This problem has been proved to be NP-hard; see [15] and [4, 5, 18] for further hardness and approximability results.

Borndörfer et al. [1, 2] investigated a problem similar to our variant of non-adaptive followers and fixed ticket prices. In contrast to our model, they assume the cost of the followers to be given by the *sum* of travel costs and probabilities, reducing the followers minimization problem to a standard shortest path problem. Based on this simplification, the authors derive a compact mixed integer program (MIP) for the leader’s optimization problem and solve it on several test instances derived from the German motorway network. In addition, they also study the case where the leader wants to minimize the number of fare evaders. In contrast, we model the followers’ response more precisely by considering the exact expected fine along a path, leading to a nonlinear aggregation of probabilities. In our computational study, we systematically compare the difference between both modeling approaches. We show that the quality of plain MIP solutions (as in [1, 2]) is on average over all instances 5% lower than our solutions when taking the exact followers response into account. For instances derived from the Amsterdam metro network, our solution improves even by 7.4% on average, with an improvement of more than 20% on some instances.

There is a body of literature on so-called network security games, where a defender can check a limited number of edges in a graph and the attacker

computes from a given set of source and target nodes a path connecting a source and a target; see [14,20,23]. The payoff for the attacker (and defender) has a binary characteristic since it only depends on whether or not the attacker has been caught. In Lin et al. [16], a single defender moves through the network *over time* to detect attackers. The authors propose a linear program which solves the problem to optimality for small instances, and heuristics which solve the problem for larger instances within 1% of optimality.

Yin et al. [24] study a fare inspection model similar to ours. In contrast to our work, passengers are assumed to follow a fixed route through the network. In [3], the problem of computing tours of inspectors for a given distribution of traffic flows is considered. The authors derive an integer programming formulation and conduct a computational study on the German motorway network.

2 The Model

We are given a directed graph $G = (V, E)$ with costs $c : E \rightarrow \mathbb{Z}_+$ modeling the transit times on the edges (as monetary cost incurred to the passengers traveling along them). In order to prevent fare evasion, the network operator (the *leader*) sets inspection probabilities $p_e \in [0, 1]$ on the edges subject to the budget constraint $\sum_{e \in E} p_e \leq B$, where $B \geq 0$ corresponds to a limited number of ticket inspectors. We first describe the passengers' reaction to the chosen inspection strategy and then discuss the resulting revenue for the operator.

The followers. Passengers (the *followers*) are modeled by a set of *commodities* K . For every commodity $i \in K$, a demand $d_i \geq 0$ is given that specifies the number of passengers with origin $s_i \in V$ and destination $t_i \in V$. Passengers of commodity i can either buy a ticket at price $T_i \geq 0$ or choose a path without paying the ticket. If a passenger is caught without a valid ticket he has to pay a fine $F \geq 0$ satisfying $F \geq T_i$. As in most public transport systems (e.g., the Dutch or German sub- and railway networks) the fine includes the ticket price and enables the passenger to continue his trip. Passengers are assumed to be rational, deciding purely based on their personal costs, which is expressed as the sum of travel times along the chosen path and the monetary cost. In the following, we differentiate between a *non-adaptive* and an *adaptive* variant of the followers response. In both settings, passengers of commodity i can either choose to pay the ticket and follow the shortest path w.r.t. c , or decide to evade the fare, choosing a path $P \in \mathcal{P}_i$ that minimizes a variant-dependent cost function (where \mathcal{P}_i denotes the set of s_i - t_i -paths).

Non-adaptive followers. In the non-adaptive variant, passengers are assumed to choose a route before the start of their trip and continue along it independent of whether or not they encounter an inspector. For a path P , we denote its total travel time by $c(P) := \sum_{e \in P} c_e$, and we denote the probability that no inspector is encountered along P by $\pi(P) := \prod_{e \in P} (1 - p_e)$. Given probabilities p , the

expected cost of path P for passenger i in the non-adaptive setting is

$$f_{N,p,i}(P) := c(P) + (1 - \pi(P)) \cdot F.$$

This variant is plausible under the assumption that passengers have determined their route beforehand (via checking the route and timetables). We denote the corresponding optimization problem by FMP_N .

Adaptive followers. In the adaptive variant, once a passenger is caught, he will continue his trip along the shortest path (w.r.t. c). Letting $\text{SP}_c(v, w)$ denote the shortest path distance from v to w w.r.t. c , the follower thus tries to find a path $P = (e_1, \dots, e_k) \in \mathcal{P}_i$ with $e_i = (v_i, v_{i+1})$ minimizing the expected cost

$$f_{A,p,i}(P) := \sum_{i=1}^k \prod_{j=1}^{i-1} (1 - p_{e_j}) \cdot \left(c_{e_i} + p_{e_i} (F + \text{SP}_c(v_{i+1}, t)) \right).$$

Note that in this formula, the i th summand corresponds to the event of v_i being reached without inspection, in which case e_i is traversed next. This variant assumes that passengers know the shortest paths from all stations to their destination in a network, e.g., by calculating it using a portable computing device. We denote the corresponding optimization problem by FMP_A .

The leader. The leader's problem can be defined as a bilevel problem, where the leader sets probabilities on the edges to which the followers respond by solving their individual optimization problems. While we will always assume the fine F to be fixed in the problem input (fines are commonly determined by the legislation [12]), we will consider both the scenario where ticket prices are *fixed* a priori and the scenario where they are *flexible* and can be determined by the leader as part of his optimization problem. Combining this with the two different models for the followers' reaction, we obtain four different versions of the *leader's maximization problem*, which we will denote by LMP_X^L , with $L \in \{\text{fix}, \text{flex}\}$ and $X \in \{N, A\}$ specifying the ticket pricing variant and the behaviour of the followers, respectively. Accordingly, we denote by $\Gamma_{X,i}^L(p)$ the leader's revenue per passenger received from commodity i when choosing inspection probabilities p . The leader thus wants to maximize $\sum_{i \in K} d_i \Gamma_{X,i}^L(p)$.

Fixed Fares. In the fixed-fare setting, the revenue received from passenger i is either the ticket price or the expected revenue from collecting fines, i.e.,

$$\Gamma_{X,i}^{\text{fix}}(p) := \max \left\{ F \cdot (1 - \pi(P)) : P^* \in \underset{P \in \mathcal{P}_i \cup \{P_i^T\}}{\text{argmin}} f_{X,p,i}(P) \right\},$$

where P_i^T denotes a special path representing the option of paying the ticket with $f_{X,p,i}(P_i^T) := \text{SP}_c(s_i, t_i) + T_i$ and $\pi(P_i^T) := 1 - T_i/F$.

Flexible Fares. When the leader is allowed to determine the ticket prices, he can ensure that every passenger chooses to buy a ticket by setting the prices sufficiently low. The maximum value a passenger is willing to pay for a ticket is

$$I_{X,i}^{\text{flex}}(p) := \min\{f_{X,p,i}(P) - \text{SP}_c(s_i, t_i) : P \in \mathcal{P}_i\},$$

because for this ticket price, the cost of paying the ticket and traveling along the shortest path is at most the expected cost of his best fare evasion option.

3 The Followers' Minimization Problem

In this section, we design efficient algorithms of both variants of the followers' minimization problem. Throughout this section, we assume we are given the graph $G = (V, E)$, the start $s \in V$ and destination $t \in V$ of a given follower, costs c , probabilities p , and the fine F .

Non-adaptive Followers' Minimization Problem. The non-adaptive version of the followers' minimization problem is related to the *restricted shortest path problem* (RSP): If P^* is an s - t -path minimizing f_N , then P^* also maximizes $\pi(P^*)$ among all paths with $c(P) \leq C^* := c(P^*)$. By discretizing the set of possible values for C^* , this relation can be used to derive an FPTAS for FMP_N ; see Appendix A.1 for a detailed proof.

Theorem 1. *There is an algorithm for FMP_N that computes a $(1+\varepsilon)$ -approximate solution in time polynomial in $\frac{1}{\varepsilon}$, $\log c_{\max}$, $|V|$, and $|E|$.*

It is well-known that RSP is NP -hard even on a very restricted subclass of series-parallel graphs. Given the similarity of the two problems, it seems natural to expect this hardness result to carry over to FMP_N . Surprisingly, it turns out that FMP_N can be solved exactly in polynomial time on series-parallel graphs.

Theorem 2. *There is an algorithm that computes an optimal solution to instances of FMP_N with series-parallel graphs in polynomial time.*

Sketch of proof. The algorithm makes use of the inductive definition of a series-parallel graph as the series or parallel composition of two smaller series-parallel graphs. It recursively decomposes the graph and computes for each of the two subgraphs a set of candidate subpaths through them. It can be shown that an optimal path through the entire graph can be obtained from a composition of those candidate subpaths. By carefully bounding the probability of an inspection on the entire path depending on the parameters of the corresponding subpaths, we can restrict ourselves to a polynomial number of combinations of subpaths. A complete description and analysis of the algorithm can be found in Appendix A.2.

An exact algorithm for general graphs. We also provide an exact, however non-polynomial, algorithm for FMP_N on general graphs. The algorithm is based on a bidirectional vertex labeling approach and enumerating all Pareto-optimal paths, where a path P is dominated by a path P' if both $c(P) \geq c(P')$ and $\pi(P) \leq \pi(P')$. The algorithm is described in detail in Appendix A.3. In order to speed up the enumeration, we use an adaptation of *contraction hierarchies* [11], a preprocessing technique that performs node contractions, enabling a more efficient traversal of the paths in the graph. This technique is known to work especially well on sparse graphs, a common property of transit networks.

Adaptive Followers' Minimization Problem. The adaptive version of the followers minimization problem can be solved in polynomial time using a label setting algorithm. The algorithm makes use of the following observation on the structure of the cost function f_A , with $P[v, w]$ denoting the v - w -subpath of P .

Lemma 1. $f_A(P) = f_A(P[s, v]) + \pi(P[s, v]) \cdot f_A(P[v, t])$.

As an immediate consequence of Lemma 1, we can deduce that every suffix of an optimal path must also be an optimal path.⁴

Corollary 1. *Let P^* be an s - t -path minimizing f_A and let $v \in V(P^*)$. Then $P^*[v, t]$ is a v - t -path minimizing f_A .*

In the spirit of Dijkstra's shortest path algorithm [9], our algorithm iteratively computes the cost of an optimal v - t -path for some vertex v . It maintains the set S of vertices for which an optimal path has been computed and a value $\phi(v)$ for every vertex $v \in V$, denoting the cost of the cheapest v - t -path found so far. In every iteration, a vertex $w \in V \setminus S$ with minimum value $\phi(w)$ is added to S and the labels $\phi(v)$ of vertices v with $(v, w) \in E$ are updated accordingly. The complete algorithm and its analysis can be found in Appendix A.4.

Theorem 3. *There is an algorithm that solves FMP_A in time $O(|E| + |V| \log |V|)$.*

The Impact of Adaptivity. We now prove a tight upper bound of $4/3$ on the ratio of the optimal cost between non-adaptive and adaptive strategies.

Theorem 4. *Let OPT_N be the cost of an optimal non-adaptive solution and OPT_A be the cost of an optimal adaptive solution. Then $\text{OPT}_N \leq \frac{4}{3} \text{OPT}_A$.*

Proof. Let P be a path that minimizes f_A . Observe that

$$\text{OPT}_A = f_A(P) \geq \pi(P)c(P) + (1 - \pi(P))(\text{SP}_c(s, t) + F),$$

where the inequality is due to the fact that in any case, the passenger has at least to traverse a shortest path from s to t . Furthermore,

$$\text{OPT}_N \leq \min \{c(P) + (1 - \pi(P))F, \text{SP}_c(s, t) + F\}$$

⁴ Note however that the same is not true for prefixes of the optimal path, as the arcs of $P[s, v]$ also appear in the second summand.

as both P and a shortest path from s to t are feasible non-adaptive solutions. Thus,

$$\frac{\text{OPT}_N}{\text{OPT}_A} \leq \max_{\substack{p \in [0,1], F \geq 0 \\ 0 \leq S \leq C}} \frac{\min\{C + pF, S + F\}}{(1-p)C + p(S + F)} \leq \max_{p \in [0,1]} \frac{1}{1-p+p^2} \leq \frac{4}{3},$$

where the second inequality comes from the fact that the maximum is attained for $C + pF = S + F$. \square

The following example shows that the bound given in Theorem 4 is tight.

Example 1. Let $V = \{s, v, t\}$ and $E = \{e_0, e_1, e_2\}$ where e_0 goes from s to v and e_1 and e_2 are two parallel arcs from v to t . The travel costs are $c_{e_0} = c_{e_1} = 0$ and $c_{e_2} = 1$, the inspection probabilities are $p_{e_0} = 1/2$, $p_{e_1} = 1$, and $p_{e_2} = 0$, and $F = 2$. Observe that there are only two s - t -paths $P_1 = (e_0, e_1)$ and $P_2 = (e_0, e_2)$ and that $f_N(P_1) = f_N(P_2) = 2$ and $f_A(P_2) = \frac{3}{2}$, yielding a ratio of $\frac{4}{3}$ between optimal non-adaptive and adaptive strategies.

Remark 1. Note that in the proof of Theorem 4, we did not make use of the fact that the probability $\pi(P)$ is determined by individual probabilities on the arcs of the network. Therefore, the bound of $\frac{4}{3}$ given by the theorem is still true for arbitrary probabilities $\pi(P)$ for each path P . In particular, the result still holds if inspections at specific edges are not independent events.

4 The Leader's Maximization Problem

In this section, we discuss algorithms and complexity results for the leader's maximization problem. On the theoretical side, we derive NP -hardness for a restricted special case of the problem and an LP relaxation, which yields upper bounds on the profit for all four model variants. For the flexible-fare setting, we also obtain a constant factor approximation. On the practical side, we propose a local search procedure that, combined with initial solutions from the LP relaxation computes close-to-optimal solutions for all four variants of the problem.

Complexity of the Leader's Problem. The following NP -hardness result can be derived by a simple reduction from the directed multicut problem; see Appendix B.1.

Theorem 5. LMP_X^L for $L \in \{\text{fix}, \text{flex}\}$ and $X \in \{A, N\}$ is strongly NP -hard, even when restricted to instances with $|K| = 2$ and $c \equiv 0$.

LP Relaxation. Let OPT_X^L denote the value of an optimal solution to the corresponding version of the leaders maximization problem for $L \in \{\text{flex}, \text{fix}\}$ and $X \in \{A, N\}$. The following lemmas relate these values to one another.

Lemma 2. $\text{OPT}_N^{\text{flex}} \geq \text{OPT}_N^{\text{fix}}$ and $\text{OPT}_A^{\text{flex}} \geq \text{OPT}_A^{\text{fix}}$.

Lemma 3. $\frac{4}{3}\text{OPT}_A^{\text{flex}} \geq \text{OPT}_N^{\text{flex}} \geq \text{OPT}_A^{\text{flex}}.$

In order to obtain the LP relaxation, we will make use of a linearization approach, which is based on the following classic approximation.

Lemma 4. $1 - \pi(P) \leq \min \{ \sum_{e \in P} p_e, 1 \} \leq \frac{1}{1-e^{-1}} (1 - \pi(P)).$

Using Lemma 4, we replace the term $1 - \pi(P)$ in the followers' objective function by $\sum_{e \in P} p_e$. Note that after this replacement, the non-adaptive version of FMP corresponds to a classic shortest path problem. Using the dual of the shortest path LP, we derive the following LP relaxation for $\text{LMP}_N^{\text{flex}}$.

$$\begin{aligned}
[\text{LP}] \quad & \max \quad \sum_{i \in K} d_i(y_i(t_i) - y_i(s_i) - \text{SP}_c(s_i, t_i)) \\
\text{s.t.} \quad & \sum_{e \in E} p_e \leq B \\
& y_i(w) - y_i(v) \leq c_e + Fp_e \quad \forall i \in K, e = (v, w) \in E \\
& p_e \in [0, 1] \quad \forall e \in E
\end{aligned}$$

The value OPT_{LP} of an optimal solution to [LP] yields an upper bound to all four variants of the leader's maximization problem.

Lemma 5. $\text{OPT}_{\text{LP}} \geq \text{OPT}_X^L$ for all $L \in \{\text{flex}, \text{fix}\}$ and $X \in \{A, N\}$.

Using Lemmas 3 and 4, we can also derive that using an optimal solution to [LP] yields approximation algorithms for $\text{LMP}_N^{\text{flex}}$ and $\text{LMP}_A^{\text{flex}}$.

Theorem 6. *There is a $(1 - 1/e)$ -approximation algorithm for $\text{LMP}_N^{\text{flex}}$.*

Corollary 2. *There is a $\frac{3}{4}(1 - 1/e)$ -approximation algorithm for $\text{LMP}_A^{\text{flex}}$.*

The analysis of the algorithm given in Theorem 6 is tight. A corresponding instance is described in Appendix B.2, along with the proofs of the preceding results. There, it is also shown that [LP] does not yield any constant approximation guarantees for the fixed-fare setting.

Local Search Framework. We conclude this section by presenting a general local search framework to compute close-to-optimal solutions to the leader's maximization problem. The approach can be applied to all four model variants by using the corresponding followers' response and leader's objective function.

The Algorithm. In addition to an instance of LMP_X^L , the input for the algorithm consists of a candidate subset $S \subseteq E$ of the graph edges and an initial setting of probabilities on these edges. An improving move of the local search chooses two disjoint subsets $E^+, E^- \subset S$ with $|E^+|, |E^-| \leq k$ for an input parameter k . The probabilities on the edges in E^- are uniformly decreased up to a total decrease of $\Delta' = \min\{\Delta, \sum_{e \in E^-} p_e, \sum_{e \in E^+} 1 - p_e\}$, where $\Delta > 0$ is an exponentially

decreasing step size. Then, the probabilities on the edges in E^+ are uniformly increased up to a total increase of Δ' . The framework then recomputes the followers' response, and accepts the move, if the leader's profit increases or reverts it, otherwise. The algorithm terminates, if the improvement in the objective is below a given threshold for a given number of consecutive iterations.

Initial Solutions. The performance of the local search framework depends crucially on the choice of candidate edges and the corresponding initial solution. We tested several methods for generating such solutions. The first is solving the LP relaxation [LP] and using the support of the resulting solutions as candidate edge set. The second is computing a minimum cardinality directed multicut in the graph separating all terminal pairs, and then distributing the budget uniformly among the arcs in the cut. In addition, we also used the solutions from the MIP formulation given in [1] for the fixed price setting.

5 Computational Study

In this section, we present an extensive computational study on a broad set of realistic instances, assessing the solution quality of our local search approach and the impact of our new modelling approach compared to existing models. See Appendix C for a complete overview of our computational results and details of implementation and generation of instances.

Test Instances. Experiments were performed on two different instance sets. The first set comprises the complete networks of the Amsterdam subway system and the Dutch railway system, as well as a subnetwork of the latter restricted to major transport hubs. The data was acquired from Dutch Railways [17]. Due to privacy regulations, no real-world passenger data was available. Therefore, for each network, we generated ten instances each with 25, 50, 100, and 200 commodities by choosing pairs of vertices uniformly at random, and drawing the corresponding demands uniformly at random from the interval $[1, 50]$. Ticket prices were determined using a linearized variant of the formula used in official regulations, and travel costs (in monetary units) were calculated from actual travel times with a conversion rate of 0.132 Euro per minute travel time; see [17]. The second instance set comprises randomly generated planar graphs exhibiting characteristics similar to those of real-world networks [22]. The graphs were generated using an approach similar to the one in [8]. We generated ten graphs for each possible combination of $|V|, |K| \in \{25, 50, 100, 200\}$. For all instances, we tested 20 different values of budgets from the range of 0.2 to 25. In total this yields 800 instances for each of three real-world networks and each of the four graph size classes of the randomized set, yielding 5600 instances in total.

Algorithms. As algorithms for computing start solutions of the local search, we tested the LP relaxation (**LP**) and a minimum cardinality multi-cut computed using a standard MIP formulation (**MC**). In order to assess the impact

graph set	$\text{LMP}_N^{\text{fix}}$				$\text{LMP}_A^{\text{fix}}$			$\text{LMP}_N^{\text{flex}}$		$\text{LMP}_A^{\text{flex}}$	
	best	LP	LP^{LS}	Δ_{MIP}	best	LP	LP^{LS}	LP	LP^{LS}	LP	LP^{LS}
nlmajor	97.4	94.8	97.3	2.09	97.4	94.1	97.2	96.4	98.0	96.4	98.0
nlcomplete	94.6	91.6	93.2	1.28	93.8	90.9	92.6	97.8	97.9	97.7	97.9
adammetro	98.2	94.7	97.7	7.09	98.2	94.7	97.8	94.7	97.8	94.7	97.8
small	97.0	92.1	96.3	4.54	96.8	88.1	96.0	95.3	97.5	95.3	97.4
medium	96.3	90.7	95.2	4.79	96.0	88.9	95.0	95.5	97.5	95.4	97.3
large	95.7	90.4	95.1	4.82	95.3	88.6	94.5	96.6	98.0	96.5	97.8
huge	95.6	89.1	94.1	4.03	94.6	85.2	92.7	96.5	97.9	96.4	97.7

Table 1: Average ratios between solutions and upper bounds in percent. ‘best’ denotes the average over the best solutions found for each instance, LP denotes the solutions found by [LP], LP^{LS} denotes the solution found by performing the local search heuristic on the LP solution, and Δ_{MIP} denotes the improvement of ‘best’ compared to the solution found by using the MIP from [1].

of the more precise followers’ objective function in our model as compared to existing approaches, we additionally computed the mixed integer programming solutions from [1] (MIP). The follower’s response in the local search procedure was computed using the exact algorithms for the respective variants presented in Section 3. After initial experiments for fine-tuning the parameters of the framework, it turned out that restricting to $k = 1$, i.e., probability shifts from one edge to another, is already sufficient for obtaining close-to-optimal solutions within 30 iterations. We also set the initial step length Δ to 0.1, decreasing it by a factor of 0.9 in every iteration.

Solution Quality. Table 1 and Figure 1 show average gaps for all models on the test instances. With the exception of few instances with very high budget, results from LP^{LS} consistently dominated those of MC^{LS} . We therefore omitted stating the results of the latter. Solutions for the fixed-fare variant are within 95 % of the upper bounds on average, while solutions for the flexible-fare variant are within 97.5 %. This slight difference can be explained by the fact that the same upper bound was used for all four variants.

Impact of the Budget Size. We also investigated the impact of budget sizes on the optimality gap and the achieved profit; see Figure 2 in Appendix C for a visualization. In the range of $B \in [1, 4]$, the ratio of obtained profit and upper bound exhibits a bath tub curve behavior until it stabilizes at a ratio of 1 for larger budgets (which is expected as for a large enough budget, all passengers can be forced to buy a ticket). The achievable profit as a function of the budget is concave in all investigated examples, which might be a universal property of the optimal value function (as a function of the budget).

Comparison with Existing Models. For the fixed-fare variant with non-adaptive followers, we can compare our modelling approach to that of [1]. We

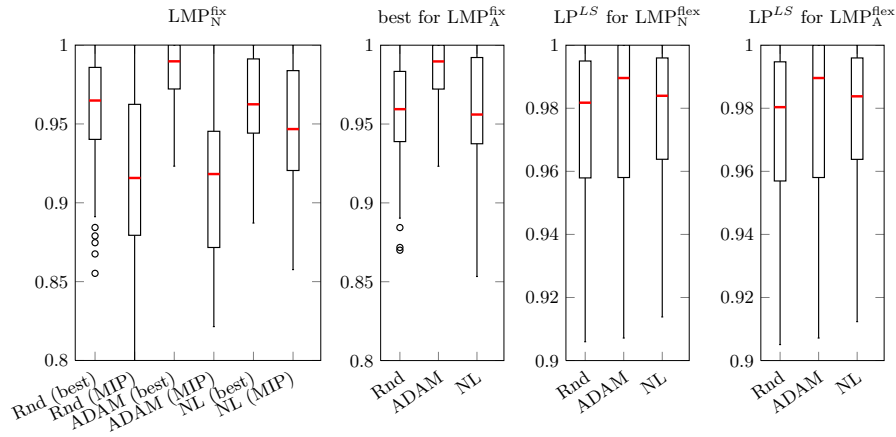


Fig. 1: Ratio of profits to upper bounds. The diagrams show the distribution of the ratios for instances in the respective test sets. The mark inside each box denotes the median, boxes represent lower and upper quartiles and the whisker ends show the minimum and maximum, respectively, apart from possible outliers marked by a cycle. The diagrams were plotted following the suggestions in [10].

solved the MIP formulation proposed in [1] and computed the leader’s profit resulting from the realistic response of followers (i.e., without linearization of their objective function). Comparing these solutions to the ones derived from our local search procedure, we observed that our approach yields an increase in profit of about 5% on average on the randomly generated instances, about 7.5% on the metro, and about 2% on the railway network. In fact for some of the metro instances, the increase exceeds 20%; see Table 1 and Figure 1.

Acknowledgements. We thank Ralf Borndörfer and Elmar Swarat for giving an inspiring talk on the topic of toll enforcement that motivated our study.

References

1. R. Borndörfer, J. Buwaya, G. Sagnol, and E. Swarat. Optimizing toll enforcement in transportation networks: a game-theoretic approach. *Electronic Notes in Discrete Mathematics*, 41:253–260, 2013.
2. R. Borndörfer, B. Omont, G. Sagnol, and E. Swarat. A Stackelberg game to optimize the distribution of controls in transportation networks. In *Proceedings of the 3rd International Conference on Game Theory for Networks*, 2012.
3. R. Borndörfer, G. Sagnol, and E. Swarat. A case study on optimizing toll enforcements on motorways. In *Proceedings of the 2nd Student Conference on Operational Research*, 2012.
4. M. Bouhtou, A. Grigoriev, S. van Hoesel, A. F. van der Kraaij, F. C.R. Spieksma, and M. Uetz. Pricing bridges to cross a river. *Naval Research Logistics*, 54(4):411–420, 2007.

5. P. Briest, M. Hoefer, and P. Krysta. Stackelberg network pricing games. *Algorithmica*, 62(3-4):733–753, 2012.
6. L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard. A bilevel model for toll optimization on a multicommodity transportation network. *Transportation Sci.*, 35(4):345–358, 2001.
7. L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard. Joint design and pricing on a network. *Oper. Res.*, 56(5):1104–1115, 2008.
8. A. Denise, M. Vasconcellos, and D. J. A. Welsh. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.
9. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
10. M. Frigge, D.C. Hoaglin, and B. Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.
11. R. Geisberger, P. Sanders, D. Schultes, and C. Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Sci.*, 46(3):388–404, 2012.
12. B. C. M. Gijsbers. Directeur Openbaar Vervoer en Spoor, Private communication.
13. R. Hassin. Approximation schemes for the restricted shortest path problem. *Math. Oper. Res.*, 17(1):36–42, February 1992.
14. M. Jain, V. Conitzer, and M. Tambe. Security scheduling for real-world networks. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, 2013.
15. M. Labbé, P. Marcotte, and G. Savard. A bilevel model of taxation and its application to optimal highway pricing. *Man. Sci.*, 44(12):1608–1622, 1998.
16. K. Y. Lin, M. P. Atkinson, T. H. Chung, and K. D. Glazebrook. A graph patrol problem with random attack times. *Oper. Res.*, 61(3):694–710, 2013.
17. NederlandseSpoorwegen (Dutch Railways). NS tarieven (Dutch tariff regulations). <http://www.ns.nl/reizigers/klantenservice/klantenservice/tarieven>, April 2014.
18. S. Roch, G. Savard, and P. Marcotte. An approximation algorithm for Stackelberg network pricing. *Networks*, 46(1):57–67, 2005.
19. Subsecretaría de Transportes, Gobierno de Chile. Fare evasion in Transantiago. <http://www.fiscalizacion.cl/index.php/evasion-de-transantiago-alcanza-192-y-registra-nivel-mas-bajo-desde-diciembre-de-2010/>, April 2014.
20. M. Tambe. *Security and game theory: Algorithms, deployed systems, lessons learned*. Cambridge University Press, 2012.
21. Transport for London. Bus fare evasion communications development. Technical report, Transport for London, 2010.
22. C. Von Ferber, T. Holovatch, Y. Holovatch, and V. Palchykov. Public transport networks: Empirical analysis and modeling. *The European Physical Journal B*, 68(2):261–275, 2009.
23. A. Washburn and K. Wood. Two-person zero-sum games for network interdiction. *Oper. Res.*, 43(2):243–251, 1995.
24. Z. Yin, A. X. Jiang, M.P. Johnson, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, M. Tambe, and J.P. Sullivan. TRUSTS: Scheduling Randomized Patrols for Fare Inspection in Transit Systems. In *Proceedings of the 24th Conference on Innovative Applications of Artificial Intelligence*, 2012.

A Appendix for the Followers' Minimization Problem (Section 3)

A.1 An FPTAS for FMP_N (Proof of Theorem 1)

Theorem 1. *There is an algorithm for FMP_N that computes a $(1+\varepsilon)$ -approximate solution in time polynomial in $\frac{1}{\varepsilon}$, $\log c_{\max}$, $|V|$, and $|E|$.*

Proof. Define $c_{\max} := \max_{e \in E} c_e$, $k := \lceil \log_{1+\varepsilon} |V| c_{\max} \rceil$, and $C_i := (1 + \varepsilon)^i$ for $i \in [k]$. For each $i \in [k]$, we use the restricted shortest path algorithm by Hassin [13] to compute a path P_i with $c(P_i) \leq (1 + \varepsilon)C_i$ and $\pi(P_i) \geq \pi(P)$ for all P with $c(P) \leq C_i$. Choose a path P' from P_0, \dots, P_k minimizing $f_N(P')$ among all s_i - t_i -paths of the graph. Let P^* be an s - t -path minimizing f_N and define $i^* := \min\{i \in [k] : c(P^*) \leq C_i\}$. Then $c(P_{i^*}) \leq (1 + \varepsilon)^2 c(P^*)$ and $\pi(P_{i^*}) \geq \pi(P^*)$ and thus $f_N(P') \leq f_N(P_{i^*}) \leq (1 + \varepsilon)^2 f_N(P^*)$. \square

A.2 An Exact Polynomial Time Algorithm for FMP_N on Series-Parallel Graphs (Proof of Theorem 2)

Our algorithm makes use of the inductive definition of a series-parallel graph as the series or parallel composition of two smaller series-parallel graphs. It recursively decomposes the graph and computes for each of the two smaller graphs a set of paths together with corresponding intervals. These paths and intervals will fulfil the following property: If an optimal path P^* traverses the subgraph G' , then $\pi(P^*)$ must be in one of the intervals of the subgraph and the subpath of P^* in G' must have the same cost and inspection probability as the path corresponding to the interval. We can thus replace this part of P^* by the corresponding path without changing the objective function value. This implies that for the original graph G , one of the polynomially many paths computed by the procedure must be optimal.

We start with a simple observation, which is true for FMP_N in any graph and will be useful in our algorithm.

Lemma A1. *Let P_1 and P_2 be two s - t -paths and let $v, w \in V(P_1) \cap V(P_2)$. For $i \in \{1, 2\}$ define $c_i := c(P_i[v, w])$ and $\pi_i := \pi(P_i[v, w])$. Assume $c_1 > c_2$ and $\pi_1 > \pi_2$.*

- If P_1 minimizes f_N among all s - t -paths, then $\pi(P_1) \geq \frac{c_1 - c_2}{\pi_1 - \pi_2} \cdot \frac{\pi_1}{F}$.
- If P_2 minimizes f_N among all s - t -paths, then $\frac{c_1 - c_2}{\pi_1 - \pi_2} \cdot \frac{\pi_2}{F} \geq \pi(P_2)$.

Proof. Consider the path $P' := P_1[s, v] \circ P_2[v, w] \circ P_1[w, t]$. As P_1 minimizes f_N we have

$$0 \leq f_N(P') - f_N(P_1) = c_2 - c_1 + (1 - \frac{\pi_2}{\pi_1})\pi(P_1)F.$$

This immediately implies the claimed lower bound on $\pi(P_1)$. The upper bound for $\pi(P_2)$ follows analogously. \square

Algorithm 1 Algorithm for FMP_N on Series-parallel Graphs

 $(\mathcal{P}, \ell, u) \leftarrow \text{FINDPATHS}(G, s, t, c, p, F)$ Choose $P' \in \text{argmin}_{P \in \mathcal{P}} f_N(P)$.**return** P' **procedure** $\text{FINDPATHS}(G = (V, E), s, t, c, p, F)$ **if** $E = \{(s, t)\}$ **then** $P \leftarrow (s, t); \ell(P) = 0; u(P) = 1$ $\mathcal{P} = \{P\}$ **else if** G is series composition of G_1 and G_2 joined at v **then** $(\mathcal{P}_1, \ell_1, u_1) \leftarrow \text{FINDPATHS}(G_1, s, v, c, p, F)$ $(\mathcal{P}_2, \ell_2, u_2) \leftarrow \text{FINDPATHS}(G_2, v, t, c, p, F)$ $\mathcal{P} \leftarrow \emptyset$ **for each** $P_1 \in \mathcal{P}_1, P_2 \in \mathcal{P}_2$ with $[\ell_1(P_1), u_1(P_1)] \cap [\ell_2(P_2), u_2(P_2)] \neq \emptyset$ **do** $P \leftarrow P_1 \circ P_2$ $\ell(P) \leftarrow \max\{\ell_1(P_1), \ell_2(P_2)\}$ $u(P) \leftarrow \min\{u_1(P_1), u_2(P_2)\}$ $\mathcal{P} \leftarrow \mathcal{P} \cup \{P\}$ **end for** **else if** G is parallel composition of G_1 and G_2 **then** $(\mathcal{P}_1, \ell_1, u_1) \leftarrow \text{FINDPATHS}(G_1, s, t, c, p, F)$ $(\mathcal{P}_2, \ell_2, u_2) \leftarrow \text{FINDPATHS}(G_2, s, t, c, p, F)$ $\mathcal{P} \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2$ **for each** $P \in \mathcal{P}$ **do** $\ell(P) \leftarrow 0; u(P) \leftarrow 1$ **for each** $P' \in \mathcal{P} \setminus \{P\}$ **do** **if** $c(P) \leq c(P')$ **then** **if** $\pi(P) \geq \pi(P')$ **then** $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P'\}$ **else if** $\pi(P) < \pi(P')$ **then** $u(P) \leftarrow \min \left\{ u(P), \frac{c(P) - c(P')}{\pi(P) - \pi(P')} \frac{\pi(P)}{F} \right\}$ **end if** **else if** $c(P) > c(P')$ **then** **if** $\pi(P) \leq \pi(P')$ **then** $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P\}$ **else if** $\pi(P) > \pi(P')$ **then** $\ell(P) \leftarrow \max \left\{ \ell(P), \frac{c(P) - c(P')}{\pi(P) - \pi(P')} \frac{\pi(P)}{F} \right\}$ **end if** **end if** **end for** **end for** **return** (\mathcal{P}, ℓ, u) **end procedure**

The formal listing of Algorithm 1 comprises the procedure `FINDPATHS`, which computes the intervals and corresponding paths by performing the series-parallel decomposition. Corresponding to this decomposition, it distinguishes three cases, which are described and analyzed in the proof of the following lemma.

Lemma A2. *Let $G' = (V', E')$ be a series-parallel subgraph of G with start vertex s' and end vertex t' . The set of paths \mathcal{P} and corresponding interval bounds ℓ and u computed by `FINDPATHS`(G', s', t', c, p, F) fulfill the following properties.*

- $|\mathcal{P}| \leq |E'|$
- $[\ell(P), u(P)] \cap [\ell(P'), u(P')] = \emptyset$ for $P \neq P'$
- If P^* is an s - t -path in G minimizing f_N with $s', t' \in V(P^*)$ and $P^*[s', t'] \subseteq E'$, then there is an s' - t' -path $P \in \mathcal{P}$ with $\pi(P^*) \in [\ell(P), u(P)]$, $c(P^*[s', t']) = c(P)$, and $\pi(P^*[s', t']) = \pi(P)$.

Proof. We prove the lemma by induction on the algorithm.

- Case 1: G' is only the single edge (s', t') . In this case, `FINDPATHS` returns the corresponding single-edge path and the statement of the lemma is trivially fulfilled.
- Case 2: G' is the series composition of two series-parallel graphs G_1 and G_2 joined at a vertex v . Let $(\mathcal{P}_1, \ell_1, u_1)$ and $(\mathcal{P}_2, \ell_2, u_2)$ be the sets of paths and interval bounds computed by `FINDPATHS` for G_1 and G_2 , respectively. Every path $P \in \mathcal{P}$ in the set computed for G' corresponds to the concatenation of two paths $P_1 \in \mathcal{P}_1$ and $P_2 \in \mathcal{P}_2$ such that the corresponding intervals $[\ell_1(P_1), u_1(P_1)]$ and $[\ell_2(P_2), u_2(P_2)]$ intersect. Every intersection starts either with a value $\ell_1(P_1)$ or a value $\ell_2(P_2)$. As the intervals defined by ℓ_1 and u_1 (ℓ_2 and u_2 , respectively) are disjoint, every value of $\ell_2(P_2)$ (of $\ell_1(P_1)$, respectively) occurs at most in one intersection. Therefore $|\mathcal{P}| \leq |\mathcal{P}_1| + |\mathcal{P}_2| \leq |E'|$.

If P^* traverses G' , it also has to traverse both G_1 and G_2 . By induction hypothesis, there is an s' - v -path $P_1 \in \mathcal{P}_1$ such that $P^*[s', v] \in [\ell_1(P_1), u_1(P_1)]$, $c(P^*[s', v]) = c(P_1)$, and $\pi(P^*[s', v]) = \pi(P_1)$. Furthermore, there is a v - t' -path $P_2 \in \mathcal{P}_2$ such that $P^*[v, t'] \in [\ell_2(P_2), u_2(P_2)]$, $c(P^*[v, t']) = c(P_2)$, and $\pi(P^*[v, t']) = \pi(P_2)$. This implies $[\ell_1(P_1), u_1(P_1)] \cap [\ell_2(P_2), u_2(P_2)] \neq \emptyset$ and therefore $P := P_1 \circ P_2$ is in \mathcal{P} and fulfills

$$c(P^*[s', t']) = c(P^*[s', v]) + c(P^*[v, t']) = c(P)$$

and

$$\pi(P^*[s', t']) = \pi(P^*[s', v]) \cdot \pi(P^*[v, t']) = \pi(P).$$

Therefore, the statement of the lemma is true in this case.

- Case 3: G' is the parallel composition of two series-parallel graphs G_1 and G_2 . Let $(\mathcal{P}_1, \ell_1, u_1)$ and $(\mathcal{P}_2, \ell_2, u_2)$ be the sets of paths and interval bounds computed by `FINDPATHS` for G_1 and G_2 , respectively. The set \mathcal{P} computed for G' consists of a subset of the paths from \mathcal{P}_1 and \mathcal{P}_2 . Therefore $|\mathcal{P}| \leq |\mathcal{P}_1| + |\mathcal{P}_2| \leq |E'|$. Let $P, P' \in \mathcal{P}$. Without loss of generality, we can assume

Algorithm 2 Algorithm for FMP_N on General Graphs

```
 $\mathcal{P} \leftarrow \emptyset$   
 $S \leftarrow \{(s, P_0)\}$   $\triangleright P_0$  is an empty  $s$ - $s$ -path  
while  $S \neq \emptyset$  do  
  Choose  $(v, P) \in S$  minimizing  $c(P) + F \cdot (1 - \pi(P))$ .  
   $S \leftarrow S \setminus (v, P)$   
  for each  $e = (v, w) \in \delta^+(v)$  do  
     $P' \leftarrow P \circ (e)$   
     $c(P') \leftarrow c(P) + c_e$   
     $\pi(P') \leftarrow \pi(P) \cdot (1 - p_e)$   
    if  $c(P') < c(P^*)$  or  $\pi(P') < \pi(P^*)$  for all  $(w, P^*) \in S$  then  
      if  $w = t$  then  
         $\mathcal{P} \leftarrow \mathcal{P} \cup P'$   
      else  
         $S \leftarrow S \cup \{(w, P')\}$   
      end if  
    end if  
  end for  
end while  
return  $P \in \mathcal{P}$  minimizing  $c(P) + F \cdot (1 - \pi(P))$ 
```

$c(P) > c(P')$ and $\pi(P) > \pi(P')$, as otherwise either P or P' were removed from \mathcal{P} . Therefore

$$\ell(P) \geq \frac{c(P) - c(P')}{\pi(P) - \pi(P')} \frac{\pi(P)}{F} > \frac{c(P) - c(P')}{\pi(P) - \pi(P')} \frac{\pi(P')}{F} \geq u(P')$$

by construction of u and ℓ . The corresponding intervals of P and P' are thus disjoint.

If P^* is an s - t -path minimizing f_N that traverses G' , then it either traverses G_1 or it traverses G_2 . There thus is a path $P \in \mathcal{P}_1 \cup \mathcal{P}_2$ with $c(P) = c(P^*)$ and $\pi(P) = \pi(P^*)$. Note that P is only removed from \mathcal{P} if there is a path P' with $c(P') \leq c(P)$ and $\pi(P') \geq \pi(P)$. By optimality of P^* , we must have $c(P') = c(P)$ and $\pi(P') = \pi(P)$. Thus, we can assume $P \in \mathcal{P}$ without loss of generality. It remains to show $\pi(P^*) \in [\ell(P), u(P)]$. If $\ell(P) > 0$, then there is an s' - t' -path P' in G' such that $c(P) > c(P')$ and $\pi(P) > \pi(P')$ and $\ell(P) = \frac{c(P) - c(P')}{\pi(P) - \pi(P')} \frac{\pi(P)}{F}$. Thus, Lemma A1 implies $\pi(P^*) = \pi(P) \geq \ell(P)$. Analogously, $\pi(P^*) \leq u(P)$. \square

Lemma A2 directly implies the optimality of the path computed by Algorithm 1 as well as the polynomiality of the algorithm, concluding the proof of Theorem 2.

A.3 An Exact Algorithm for FMP_N on General Graphs

We give a complete description of the exact (non-polynomial) algorithm for computing optimal solutions to FMP_N on general graphs. Our algorithm is based

on enumerating all Pareto-optimal paths. A path P is *dominated* by a path P' if both $c(P) \geq c(P')$ and $\pi(P) \leq \pi(P')$. For a vertex $v \in V$, a set of s - v -paths \mathcal{P} is a *dominating set*, if for every s - v -path P in G there is a path $P' \in \mathcal{P}$ that dominates P . Note that if P' dominates P then $f_N(P') \leq f_N(P)$. Therefore, a dominating set of s - t -paths contains an s - t -path minimizing f_N . The algorithm uses bidirectional label setting to compute a dominating set for every vertex $v \in V$. It then returns an s - t -path which minimizes f_N .

As an important pre-processing step for the algorithm, we used *contraction hierarchies* [11]. Vertices in the graph are *contracted* in a certain order, removing them while preserving the shortest paths distances between remaining vertices by adding *shortcuts*, i.e. an edge $e = (u, w)$ is added when contracting a vertex v , if u and w are adjacent to v and the shortest path from u to w is $((u, v), (v, w))$. Newly created shortcuts are added to the original graph. The sparseness of the resulting graph and the performance of the preprocessing depend crucially on the criteria used to determine the contraction order. In our setting an order that prioritizes contractions causing a large number of shortcuts turned out to be particularly efficient.

A.4 An Exact Algorithm for FMP_A (Proofs of Lemma 1 and Theorem 3)

We give a complete description of the exact polynomial time algorithm for FMP_A. We start with the proof of Lemma 1.

Lemma 1. $f_A(P) = f_A(P[s, v]) + \pi(P[s, v]) \cdot f_A(P[v, t])$.

Proof. Let $P = (e_1, \dots, e_k)$ with $e_i = (v_i, v_{i+1})$ and assume $v = v_{k'}$ for some k' with $1 \leq k' \leq k + 1$. Observe that

$$\begin{aligned}
f_A(P) &= \sum_{i=1}^k \prod_{j=1}^{i-1} (1 - p_{e_j}) \cdot \left(c_{e_i} + p_{e_i} (F + \text{SP}_c(v_{i+1}, t)) \right) \\
&= \sum_{i=1}^{k'-1} \prod_{j=1}^{i-1} (1 - p_{e_j}) \cdot \left(c_{e_i} + p_{e_i} (F + \text{SP}_c(v_{i+1}, t)) \right) \\
&\quad + \prod_{j=1}^{k'-1} (1 - p_{e_j}) \sum_{i=k'}^k \prod_{j=k'}^{i-1} (1 - p_{e_j}) \cdot \left(c_{e_i} + p_{e_i} (F + \text{SP}_c(v_{i+1}, t)) \right) \\
&= f_A(P[s, v]) + \prod_{e \in P[s, v]} (1 - p_e) \cdot f_A(P[v, t]). \quad \square
\end{aligned}$$

In the spirit of Dijkstra's shortest path algorithm [9], the algorithm iteratively computes the cost of an optimal v - t -path for some vertex v . It maintains the set S of vertices for which an optimal path has been computed and a value $\phi(v)$ for every vertex $v \in V$, denoting the cost of the cheapest v - t -path found so far. In every iteration, a vertex $w \in V \setminus S$ with minimum value $\phi(w)$ is added to S and

Algorithm 3 Algorithm for FMP_A

```

 $\phi(t) \leftarrow 0$ 
 $\phi(v) \leftarrow \infty$  for all  $v \in V \setminus \{t\}$ 
 $S \leftarrow \emptyset$ 
while  $s \notin S$  do
    Choose  $w \in V \setminus S$  minimizing  $\phi$ .
     $S \leftarrow S \cup \{w\}$ 
    for each  $e = (v, w) \in \delta^-(w)$  do
         $\phi' \leftarrow c_e + p_e(\text{SP}_c(w) + F) + (1 - p_e)\phi(w)$ 
        if  $\phi' < \phi(v)$  do
             $\phi(v) \leftarrow \phi'$ 
             $\text{next}(v) \leftarrow e$ 
        end if
    end for each
end while
 $w \leftarrow s$ ;  $P \leftarrow \emptyset$ 
while  $w \neq t$  do
     $e \leftarrow \text{next}(w)$ ;  $w \leftarrow \text{head}(e)$ 
    Add  $e$  to  $P$ .
end while
return  $P$ 

```

the labels $\phi(v)$ of vertices v with $(v, w) \in E$ are updated if a v - t -path consisting of (v, w) and an optimal w - t -path is cheaper than the current value of $\phi(v)$. A complete listing is given as Algorithm 3.

Lemma A3. *When vertex $v \in V$ is added to the set S in Algorithm 3, then there is an optimal v - t -path P_v with $f_A(P_v) = \phi(v)$ starting with the edge $\text{next}(v)$.*

Proof. By contradiction assume the lemma is not true. Without loss of generality let v be the first vertex added to S that does not fulfill the statement of the lemma and consider the moment when v is added to S . Note that $v \neq t$, as t fulfills the requirements with $\phi(t) = 0$ and $P_t = \emptyset$. Let $\text{next}(v) = e = (v, w)$ for some $w \in S$. As v is the first vertex violating the lemma, there is a w - t -path P_w with $f_A(P_w) = \phi(w)$. Thus, $P_v := (e) \circ P_w$ is a v - t -path with cost

$$f_A(P_v) = c_e + p_e(\text{SP}_c(w, t) + F) + (1 - c_e)\phi(w) = \phi(v).$$

Let P' be any v - t -path, let $e' = (v', w')$ be the last edge on P' with $v' \in V \setminus S$ and $w' \in S$. Note that

$$c(P') + F \geq c_{e'} + p_{e'}(\text{SP}_c(w', t) + F) + (1 - c_{e'})\phi(w') \geq \phi(v')$$

as $\phi(w')$ denotes the cost of an optimal w' - t -path by induction hypothesis and therefore $\phi(w') \leq \text{SP}_c(w', t) + F$. Further note that $\phi(v') \geq \phi(v)$ by choice of v . In particular, this implies $\text{SP}_c(v, t) + F \geq \phi(v)$ by choosing P' as a shortest v - t -path with respect to c . Thus, for arbitrary P' again,

$$f_A(P') \geq (1 - \pi(P'[v, v'])) \cdot (\text{SP}_c(v, t) + F) + \pi(P'[v, v']) \cdot \phi(v') \geq \phi(v)$$

which proves that P_v is optimal. \square

Theorem 3. *There is an algorithm that solves FMP_A in time $O(|E| + |V| \log |V|)$.*

Proof. The optimality of the path computed by Algorithm 3 follows immediately from Lemma A3. Shortest path distances from every vertex to t can be pre-computed using Dijkstra's algorithm in time $O(|E| + |V| \log |V|)$. The remainder of the algorithm can be implemented using a Fibonacci heap for computing the vertex minimizing $\phi(v)$, guaranteeing the claimed running time. \square

B Appendix for the Leader's Maximization Problem (Section 4)

B.1 Complexity (Proof of Theorem 5)

We reduce from the *minimum directed multicut problem*. An instance of this problem is given by a directed graph $G = (V, E)$ and k commodities specified by origin-destination pairs $(s_i, t_i)_{i=1, \dots, k}$. A multicut $M \subseteq E$ is a subset of edges such that $M \cap P \neq \emptyset$ for each path $P \in \mathcal{P}_i$ and each commodity $i \in \{1, \dots, k\}$. The task is to find a multicut of minimum cardinality. The corresponding decision problem is to decide for a given graph G and an integer q whether G has a multicut of cardinality at most q . The directed multicut problem is strongly NP-hard even for $k = 2$; see Garg, Vazirani, Yannakakis: *Multicut Cuts in Directed and Node Weighted Graphs* (ICALP 2014).

Theorem 5. LMP_X^L for $L \in \{\text{fix}, \text{flex}\}$ and $X \in \{A, N\}$ is strongly NP-hard, even when restricted to instances with $|K| = 2$ and $c \equiv 0$.

Proof. Consider an instance $I = (G, (s_i, t_i)_{i=1, \dots, k}, q)$ of the directed multicut problem with $G = (V, E)$, $k = 2$, and $q \in \mathbb{Z}_+$. We will construct an instance $\hat{I} = (G, c, K, F, T, B)$ of the leader's maximization problem as follows. We introduce two commodities, one for each pair (s_i, t_i) with $i \in K := \{1, 2\}$. We set $c_e = 0$ for all $e \in E$, $B = q$ and $T_1 = T_2 = F = 1$.

We denote by OPT_X^L the value of an optimal solution to LMP_X^L . Note that since the travel costs are all zero, $f_{N,p,i} = f_{A,p,i}$ for all $i \in K$ and any setting of probabilities $p \in [0, 1]^E$. Furthermore $\Gamma_{N,i}^{\text{fix}}(p) = \min_{P \in \mathcal{P}_i} \{f_{N,p,i}(P), T_i\} = \Gamma_{N,i}^{\text{flex}}(P)$ as $\text{SP}_c(s_i, t_i) = 0$ for all $i \in K$. Therefore $\text{OPT}_N^{\text{fix}} = \text{OPT}_N^{\text{flex}} = \text{OPT}_N^{\text{flex}} = \text{OPT}_A^{\text{flex}}$.

We claim the following equivalence proving the theorem.

$$\text{OPT}_X^L \geq 2 \Leftrightarrow \text{there exists a feasible multicut of cardinality } q.$$

Suppose that there is a multicut M with $|M| \leq q$. We then define $p_e = 1$ for all $e \in M$ and $p_e = 0$ for all $e \in E \setminus M$. Note that $\sum_{e \in E} p_e = |M| \leq q$, implying that p is a feasible solution. Furthermore, every passenger encounters an inspector with probability 1 because his path has to cross the multicut. We thus obtain $\text{OPT}_X^L \geq 2$.

Conversely, assume $p \in [0, 1]^E$ is a solution with profit 2. This implies that $f_{X,i,p}(P) = 1$ for $i \in K$ and every path $P \in \mathcal{P}_i$. This is only possible if for every $P \in \mathcal{P}_i$, there is an $e \in P$ with $p_e = 1$. Therefore the set $M = \{e \in E : p_e = 1\}$ is a multicut with cardinality $|M| \leq \sum_{e \in M} p_e \leq q$. \square

B.2 The LP Relaxation (Proofs of Lemmas 2 to 5, Theorem 6, and Corollary 2)

Lemma 2. $\text{OPT}_N^{\text{flex}} \geq \text{OPT}_N^{\text{fix}}$ and $\text{OPT}_A^{\text{flex}} \geq \text{OPT}_A^{\text{fix}}$.

Proof. Let $X \in \{A, N\}$. Note that $(1 - \pi(P))F \geq f_{X,p,i}(P) - \text{SP}_c(s_i, t_i)$ for all $p \in [0, 1]^E$, $i \in K$ and any path $P \in \mathcal{P}_i$. Therefore $\text{OPT}_X^{\text{flex}} \geq \text{OPT}_X^{\text{fix}}$. \square

For the proofs of Lemma 3 and Corollary 2, we will additionally use the following stronger version of Theorem 4.

Lemma B1. Let OPT_N be the cost of an optimal non-adaptive solution and OPT_A be the cost of an optimal adaptive solution. Then $\text{OPT}_N - \text{SP}_c(s, t) \leq \frac{4}{3}(\text{OPT}_A - \text{SP}_c(s, t))$.

Proof. Let P be a path that minimizes f_A . Observe that

$$\text{OPT}_A = f_A(P) \geq \pi(P)c(P) + (1 - \pi(P))(\text{SP}_c(s, t) + F)$$

where the inequality is due to the fact that in any case, the passenger has at least to traverse a shortest path from s to t . Furthermore,

$$\text{OPT}_N \leq \min \{c(P) + (1 - \pi(P))F, \text{SP}_c(s, t) + F\}$$

as both P and a shortest path from s to t are feasible non-adaptive solutions. Thus,

$$\begin{aligned} \frac{\text{OPT}_N - \text{SP}_c(s, t)}{\text{OPT}_A - \text{SP}_c(s, t)} &\leq \max_{\substack{p \in [0, 1], F \geq 0 \\ 0 \leq S \leq C}} \frac{\min \{C + pF, S + F\} - S}{(1 - p)C + p(S + F) - S} \\ &= \max_{\substack{p \in [0, 1], F \geq 0 \\ 0 \leq S \leq C}} \frac{\min \{C - S + pF, F\}}{(1 - p)(C - S) + pF}. \end{aligned}$$

If $C - S + pF > F$, then increasing S also increases the value of the right-hand side function. If $C - S + pF < F$, then decreasing F increases the value of the right-hand side function. Thus, if the right-hand side is maximized, then $C - S + pF = F$. Therefore

$$\frac{\text{OPT}_N - \text{SP}_c(s, t)}{\text{OPT}_A - \text{SP}_c(s, t)} \leq \max_{\substack{p \in [0, 1] \\ F \geq 0}} \frac{F}{F(1 - p + p^2)} \leq \frac{4}{3},$$

which concludes the proof. \square

Lemma 3. $\frac{4}{3}\text{OPT}_A^{\text{flex}} \geq \text{OPT}_N^{\text{flex}} \geq \text{OPT}_A^{\text{fix}}$.

Proof. Let $p \in [0, 1]^E$ be any solution to the LMP instance. Note that Lemma B1 implies $\frac{4}{3}I_{A,i}^{\text{flex}}(p) \geq I_{N,i}^{\text{flex}}(p) \geq I_{A,i}^{\text{flex}}(p)$ for all $i \in K$, and therefore $\frac{4}{3}\text{OPT}_A^{\text{flex}} \geq \text{OPT}_N^{\text{flex}} \geq \text{OPT}_A^{\text{flex}}$. \square

Lemma 4. $1 - \pi(P) \leq \min \left\{ \sum_{e \in P} p_e, 1 \right\} \leq \frac{1}{1 - e^{-1}} (1 - \pi(P))$.

Proof. We prove $\pi(P) = \prod_{e \in P} (1 - p_e) \geq 1 - \sum_{e \in P} p_e$ by induction on $|P|$. This is trivial for $|P| = 1$. For $|P| > 1$ observe that for any $e' \in P$ by induction hypothesis

$$\prod_{e \in P} (1 - p_e) \geq \left(1 - \sum_{e \in P \setminus \{e'\}} p_e \right) (1 - p_{e'}) = 1 - \sum_{e \in P} p_e + p_{e'} \sum_{e \in P \setminus \{e'\}} p_e \geq 1 - \sum_{e \in P} p_e.$$

This immediately implies the first inequality stated in the lemma.

The second inequality of the lemma is trivially true if $\sum_{e \in P} p_e = 0$. Thus we assume the sum to be strictly positive without loss of generality. Define $\sigma := \min \left\{ \sum_{e \in P} p_e, 1 \right\}$ and observe that

$$\prod_{e \in P} (1 - p_e) \leq \left(1 - \frac{\sigma}{|P|} \right)^{|P|} \leq e^{-\sigma}.$$

Therefore

$$\frac{1 - \pi(P)}{\sigma} \geq \frac{1 - e^{-\sigma}}{\sigma} \geq \min_{x \in (0,1]} \frac{1 - e^{-x}}{x}.$$

The right-hand side is decreasing in x and therefore minimized for $x = 1$. \square

Lemma 5. $\text{OPT}_{\text{LP}} \geq \text{OPT}_X^L$ for all $L \in \{\text{flex}, \text{fix}\}$ and $X \in \{A, N\}$.

Proof. Let $p \in [0, 1]^E$ be an optimal solution to $\text{LMP}_N^{\text{flex}}$. For every $i \in K$, set $y_i(s_i) = 0$ and $y_i(v) = \text{SP}_{c+FP}(s_i, v)$ for all $v \in V \setminus \{s_i\}$. It is easy to check that (p, y) is a feasible solution to [LP] and that $y_i(t_i) - y_i(s_i) \geq \text{SP}_{c+pF}(s_i, t_i)$ for every $i \in K$. Therefore

$$\begin{aligned} y_i(t_i) - y_i(s_i) - \text{SP}_c(s_i, t_i) &\geq \min \left\{ \sum_{e \in P} (c_e + Fp_e) - \text{SP}_c(s_i, t_i) : P \in \mathcal{P}_i \right\} \\ &\geq \min \left\{ \sum_{e \in P} c_e + (1 - \pi(P))F - \text{SP}_c(s_i, t_i) : P \in \mathcal{P}_i \right\} \\ &= I_{N,i}^{\text{flex}}(p) \end{aligned}$$

where the second inequality follows from Lemma 4. This implies that the optimal value of the LP is at least $\text{OPT}_N^{\text{flex}}$. By Lemmas 2 and 3, $\text{OPT}_N^{\text{flex}}$ is as least as large as the optimal solution value of any of the other versions. \square

Theorem 6. *There is a $(1 - 1/e)$ -approximation algorithm for $\text{LMP}_N^{\text{flex}}$.*

Proof. Let (p, y) be an optimal solution to [LP]. Note that $y_i(t_i) - y_i(s_i) \leq \text{SP}_{c+Fp}(s_i, t_i)$ and define $\lambda_i = y_i(t_i) - y_i(s_i) - \text{SP}_c(s_i, t_i)$. Then

$$\begin{aligned} \lambda_i &\leq \min \left\{ \sum_{e \in P} (c_e + Fp_e) - \text{SP}_c(s_i, t_i), T_i \right\} \\ &\leq \sum_{e \in P} c_e - \text{SP}_c(s_i, t_i) + \min \left\{ \sum_{e \in P} p_e, 1 \right\} \cdot F \\ &\leq \frac{1}{1 - e^{-1}} \left(\sum_{e \in P} c_e - \text{SP}_c(s_i, t_i) + \left(1 - \prod_{e \in P} (1 - p_e) \right) \cdot F \right) \end{aligned}$$

for every $P \in \mathcal{P}_i$ by Lemma 4. Thus, setting the probabilities according to p yields a solution to $\text{LMP}_N^{\text{flex}}$ with profit

$$\sum_{i \in K} d_i \Gamma_{N,i}^{\text{flex}}(p) \geq (1 - 1/e) \sum_{i \in K} d_i \lambda_i \geq (1 - 1/e) \text{OPT}_N^{\text{flex}}. \quad \square$$

Tightness of the approximation factor. The analysis of the algorithm given in Theorem 6 is tight. To see this, consider the following example instance of $\text{LMP}_N^{\text{flex}}$. Let G be a directed cycle of length n , i.e., $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_n\}$ with $e_i = (v_i, v_{i+1})$ for $i \in \{1, \dots, n-1\}$ and $e_n = (v_n, v_1)$. Let K consist of n commodities with unit demand, such that $s_i = v_i$ and $t_i = v_{i-1}$ for $i \in \{2, \dots, n\}$ and $s_1 = v_1$ and $t_1 = v_n$. Note \mathcal{P}_i consists of a unique path of length $n-1$ for every $i \in K$. Finally, let $c = 0$, $F = T_i = 1$ for every $i \in K$ and $B = n/(n-1)$. Observe that the optimal solution of [LP] sets $p_e = 1/(n-1)$ for every $e \in E$. Using these probabilities as a solution to $\text{LMP}_N^{\text{flex}}$ yields a profit of $n \cdot \left(1 - (1 - \frac{1}{n-1})^{n-1}\right)$. On the other hand, setting $p_{e_1} = 1$ and $p_{e_n} = 1/(n-1)$ yields a profit of $n - 1 + 1/(n-1)$. Note that by choosing n sufficiently large the ratio between these two values can be brought arbitrarily close to $1 - 1/e$.

Optimality gap when using [LP] for fixed fares. Unfortunately, [LP] does not yield an approximation guarantee for the fixed-fare setting. To see this, consider the following instance of $\text{LMP}_X^{\text{fix}}$. There are four nodes s_1, t_1, s_2, t_2 , together with edges from s_1 to s_2 , s_2 to t_2 , and t_2 to t_1 , each with zero cost. In addition, there are L parallel edges from s_1 to t_1 , each with cost 1. Commodity 1 has origin s_1 and destination t_1 with a total demand of L . Commodity 2 has origin s_2 and destination t_2 with a total demand of 1. The budget is $1/2 + \varepsilon$ and the fine is 2. Observe that the optimal solution of the corresponding instance of [LP] sets $p_{(s_2, t_2)} = 1/2 + \varepsilon$ and $p_e = 0$ for all other edges. Interpreting these probabilities as a solution of LMPfix_X results in a profit of $1 + 2\varepsilon$, as the followers represented by commodity 1 will prefer one of the edges from s_1 to t_1 over the three-edge path. However, setting $p_{(s_2, t_2)} = 1/2$ instead yields a profit of $L + 1$.

Corollary 2. *There is a $\frac{3}{4}(1 - 1/e)$ -approximation algorithm for $\text{LMP}_A^{\text{flex}}$.*

Proof. Again, let (p, y) be an optimal solution to [LP]. By Lemma B1 and the proof of Theorem 6, $\Gamma_{A,i}^{\text{flex}}(p) \geq \frac{3}{4}\Gamma_{N,i}^{\text{flex}}(p) \geq \frac{3}{4}(1 - 1/e)(y_i(t_i) - y_i(s_i) - \text{SP}_c(s_i, t_i))$ for all $i \in K$. \square

C Appendix for the Computational Study (Section 5)

C.1 Instances and Computational Setup

Generation of Planar Graphs. The graphs were generated using an approach similar to the one in [8]. Vertices are distributed uniformly at random in the plane; iteratively, a vertex is chosen uniformly at random and connected to the closest neighbour that can be connected without violating planarity. This is repeated $3|V| - 6$ times, after which disjoint connected components are connected using nearest Euclidean neighbours in order to ensure that the entire graph is connected. All arcs are present in both directions, as common in transit networks.

Ticket Prices. For the randomly generated instances, ticket prices were set to

$$T_i = b + m \cdot \frac{\text{SP}_c(s_i, t_i)}{\max_{v,w \in V} \text{SP}_c(v, w)},$$

where b is a base price and $b + m$ the maximum ticket price allowed in the network. This linear formula is a simplification of the formula used for official regulations regarding ticket prices in public transport networks [12,17].

Instance overview. Our study consist of seven graph sets, three based on real-world transit networks and four generated using the randomization procedure described above. Each set contains 40 different combinations of graphs, customers and demands. For each of these combinations, twenty different budget values were tested, leading to 800 different instances in each graph set; see Table 2 for average graph sizes.

	nlmajor	nlcomplete	adammetro	small	medium	large	huge
$ V $	23	341	45	25	50	100	200
$ E $	60	864	88	95	195	399	806

Table 2: Average data of test instances, showing the average numbers of vertices and edges within each of the seven graph sets.

Implementation Details. All algorithms have been implemented in Java and compiled using jre7 on Windows 7 Enterprise. Computations have been performed on a machine with Intel Core 2 Duo CPU (GHz, 64 bit) and 4GB of memory using CPLEX 12.4 API for Java for the mathematical programs.

C.2 Results

Computation times. Table 3 shows average computation times for the various algorithms. Using the LP for computing start solutions, a local optimum was reached within less than a minute for most instances, with the only exception being very large graphs combined with the (non-polynomial) non-adaptive followers’ response. Furthermore, note that for large instances with many commodities, and for budgets higher than 6, the mixed integer programming formulation could not be solved in reasonable time.

graph set	LP	LP_N^{LS}	LP_A^{LS}	LPR	LPR_N^{LS}	LPR_A^{LS}	MIP	MIP_N^{LS}	MIP_A^{LS}
nlmajor	0.17	10.2	1.59	1.39	10.6	1.68	6.31	9.93	1.62
nlcomplete	7.28	931	21	75.4	873	19.7	4024	563	7.57
adammetro	0.33	6.12	1.39	1.91	9.81	2.31	3.29	11.2	2.87
small	0.04	0.75	0.11	0.29	0.8	0.11	0.6	0.77	0.11
medium	0.06	4.5	0.18	0.32	4.9	0.19	1.5	3.90	0.18
large	0.15	32.8	0.81	0.99	44.6	0.9	5.1	34.60	0.87
huge	0.60	343.1	6.9	3.9	563.2	7.8	25.5	490.0	7.50

Table 3: Average computation time in seconds for fixed fares. LP, LPR and MIP denote the solutions found by the corresponding mathematical program, and Alg_X^{LS} with $Alg \in \{LP, LPR, MIP\}$ and $X \in \{A, N\}$ denotes the solution found by performing the local search heuristic on the solution found by algorithm Alg, using the followers response X .

Influence of Budget. The leader’s achievable revenue crucially depends on the available budget. Figure 2 illustrates the quality of the solutions for the adammetro and nlcomplete instances, as well as for a small and large random instance, all with 50 commodities for the LMP_N^{fix} model, and the obtained revenue.

Complete overview of solution quality. Tables 4 to 6 show average ratios between solutions and upper bounds for all models. Here, **LPR** denotes the linear relaxation of the mixed integer program from [1] (**MIP**).

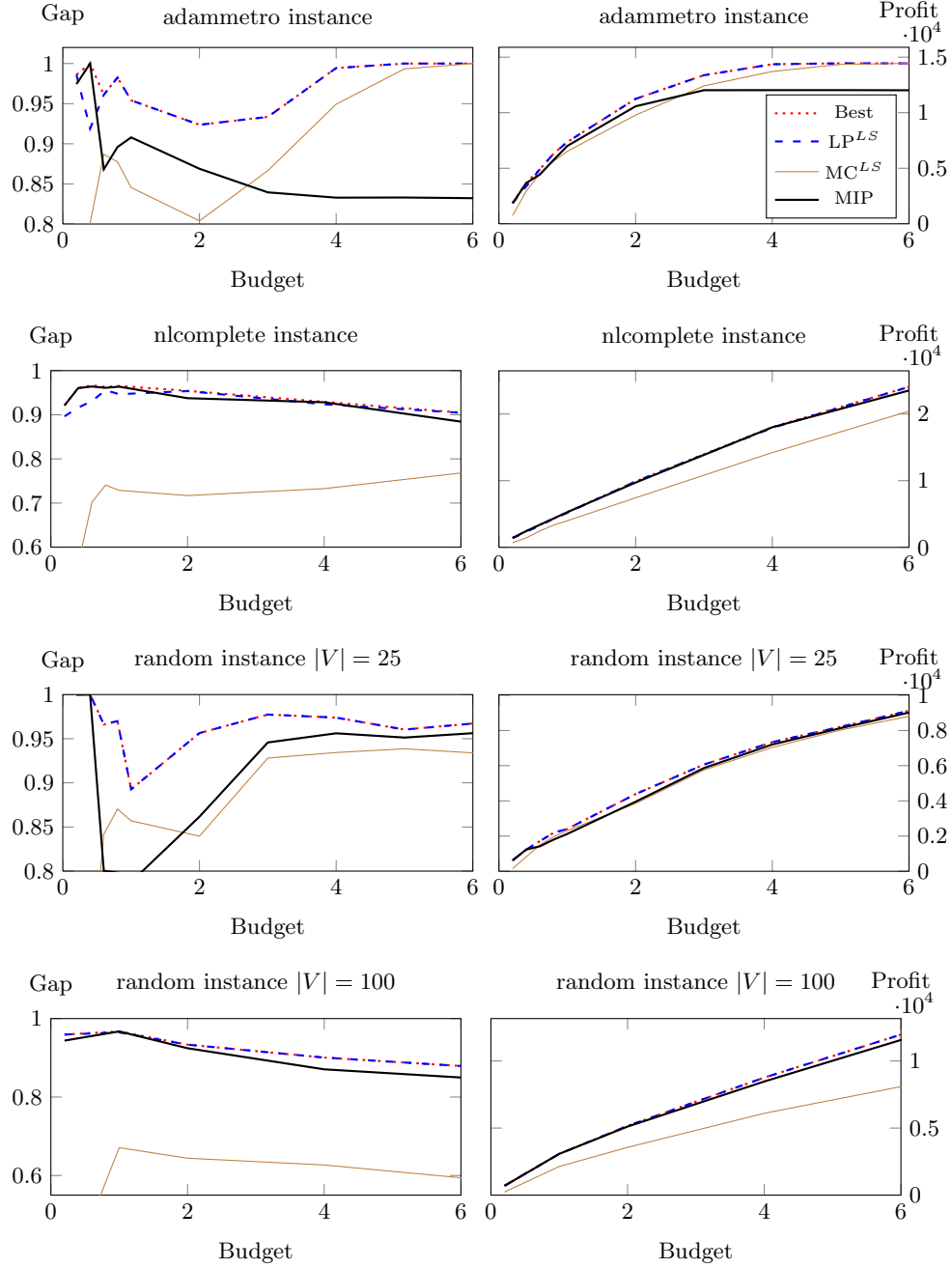


Fig. 2: Graphs illustrating the influence of the budget on the solution quality, for the adammetro instance, the nlcomplete instance, a random instance with 25 vertices and a random instance with 100 vertices, each with 50 commodities for the $\text{LMP}_N^{\text{fix}}$ model. Note that the legend in the top-right graph applies to all graphs. Graphs on the left show gaps relative to the upper bound, graphs on the right show the corresponding profit. 'Best' denotes the best solution found, LP^{LS} and MC^{LS} denote the solutions found by performing the local search heuristic on the LP solution resp. MultiCut solution, and MIP denotes the solution found by the corresponding mathematical program from [1].

graph set	$\text{LMP}_N^{\text{flex}}$				$\text{LMP}_A^{\text{flex}}$			
	LP	LP^{LS}	MC	MC^{LS}	LP	LP^{LS}	MC	MC^{LS}
nlmajor	96.4	98.0	52.8	85.2	96.4	98.0	52.8	85.2
nlcomplete	97.8	97.9	33.9	39.7	97.7	97.9	33.9	39.7
adammetro	94.7	97.8	69.0	86.7	94.7	97.8	69.0	86.7
small	95.3	97.5	49.1	77.7	95.3	97.4	49.0	77.5
medium	95.5	97.5	41.6	69.4	95.4	97.3	41.6	69.3
large	96.6	98.0	40.7	75.9	96.5	97.8	40.7	75.8
huge	96.5	97.9	37.2	71.1	96.4	97.7	37.2	71.2

Table 4: Average ratios between solutions and upper bounds in percent for test instances in the flexible-fare setting. LP^{LS} and MC^{LS} denote the solutions found by performing the local search heuristic on the LP solution and the MultiCut solution, respectively.

graph set	LP	LP^{LS}	LPR	LPR^{LS}	MIP	MIP^{LS}	MC	MC^{LS}
nlmajor	94.1	97.2	94.8	97.2	94.7	97.3	52.2	85.1
nlcomplete	90.9	92.6	90.6	92.4	91.0	93.3	33.3	37.2
adammetro	94.7	97.8	92.0	97.5	91.7	97.3	69.0	86.7
small	88.1	96.0	89.9	96.3	89.8	96.3	48.7	74.1
medium	88.9	95.0	90.1	95.0	90.5	95.1	41.1	64.9
large	88.6	94.5	89.0	94.9	89.0	95.1	40.3	71.6
huge	85.2	92.7	87.8	93.3	88.1	94.0	36.8	64.8

Table 5: Average ratios between solutions and upper bounds in percent for test instances with fixed ticket prices and adaptive followers. LP, LPR and MIP denote the solutions found by the corresponding mathematical program, and Alg^{LS} with $\text{Alg} \in \{\text{LP}, \text{LPR}, \text{MIP}\}$ denotes the solution found by performing the local search heuristic on the solution found by algorithm Alg.

graph set	LP	LP^{LS}	LPR	LPR^{LS}	MIP	MIP^{LS}	MC	MC^{LS}
nlmajor	94.8	97.3	95.1	97.3	95.4	97.4	52.2	85.1
nlcomplete	91.6	93.2	91.9	93.6	93.4	94.4	33.3	37.1
adammetro	94.7	97.8	92.0	97.5	91.7	97.3	69.0	86.7
small	92.1	96.3	92.7	96.5	92.8	96.4	48.7	74.2
medium	90.7	95.2	91.8	95.4	91.9	95.6	41.1	64.9
large	90.4	95.1	90.8	95.2	91.3	95.4	40.3	71.7
huge	89.1	94.1	91.5	94.5	91.9	95.3	36.8	64.8

Table 6: Average ratios between solutions and upper bounds in percent for test instances with fixed ticket prices and non-adaptive followers. LP, LPR and MIP denote the solutions found by the corresponding mathematical program, and Alg^{LS} with $\text{Alg} \in \{\text{LP}, \text{LPR}, \text{MIP}\}$ denotes the solution found by performing the local search heuristic on the solution found by algorithm Alg.